REMARKS

Applicants would like to thank the Examiner for the courtesies extended during the telephonic interview held December 6, 2007. Details of that interview are included herein below.

The Final Office Action dated October 3, 2007 considered claims 1-29. Claims 1-29 were rejected under 35 U.S.C. 103(a) as being unpatentable over Singhal et al. (US 7,096,418) hereinafter *Singhal* in view of Sobeski (US 6,304,879) hereinafter *Sobeski*.[1]

Applicants traverse. Applicants submit that the teachings of *Sobeski* are directed to a caching mechanism that permits one object to access static and dynamic properties of another object. However, *Sobeski* has little, if anything, to do with cache dependencies.

Figure 2 and corresponding description indicates data related to data object 204 can be temporarily stored in data cache 206. (Figure 2). Data cache 206 can store two different types of dynamic data: a) dynamic properties (e.g., text color) that are temporarily associated with data object 204 and b) state changes to static properties of data object 204 before they are committed to data object 204. (Col.1, ll. 53-58, Col. 5, ll. 19 – 29, and Col. 5, ll. 39 – 44). A data object can operate in cache mode or pass-through mode that can cause data cache 206 to be marked as a "dirty" state, which remains until data is committed. (Col. 5, ll. 51 – 58).

Data is typically committed (or not committed) to data object 204 in response to user input. (Col. 6, ll. 58 – 67). When data is actually stored to data object 204, any dynamic data in data cache 206 is lost. (Col. 7, ll. 26 – 30). Thus, data in data cache 206 is removed when data is committed to data object 204. State for any static data properties is stored to data object 204 and any dynamic properties are lost.

However, *Sobeski* fails to teach or otherwise suggest:

> an act of accessing an extensible cache dependency base class from which customized cache dependencies can be derived, the extensible cache dependency base class including a plurality of inheritable cache management methods usable by customized dependencies derived from the extensible case dependency base class;

---

[1] Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

an act of deriving a customized cache dependency class from the extensible cache dependency base class, the customized cache dependency class inheriting the plurality of inheritable cache management methods from the extensible case dependency base class, the customized cache dependency class also configured to implement further unique functionality of a customized dependency that extends the plurality of inheritable cache management methods included in the extensible cache dependency base class;

as recited in claims 1 and 28.

Sobeski also fails to teach or otherwise suggest:

an act of monitoring one or more custom dependency conditions associated with an instance of a customized cache dependency that extends a plurality of cache management methods inherited from an extensible cache dependency base class to implement unique functionality, the customized cache dependency corresponding to a customized cache dependency class that was derived from the extensible cache dependency base class;

as recited in claims 15 and 29.

Applicants submit that mere mention of an object-oriented programming environment and the use of objects to cache data does not render the claims unpatentable in view of *Singhal*. For example, data cache 206 is simply a storage location. There is no mention of data cache 206 being extendable nor being a base class. However, assuming *arguendo* that data cache 206 was an extensible base class, *Sobeski* is still absolutely silent as to how other classes can be derived from data cache 206 to extend cache functionality and how properties of data cache 206 can be inherited by other data cache objects. (Col. 5. ll. 63-67).
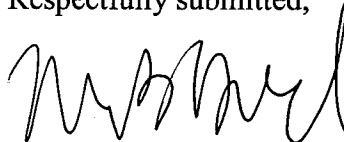
Applicants further submit that the removal of dynamic data from data cache 206 is not in fact based on any cache dependency. *Sobeski* teaches that dynamic data is removed in response to user input either committing or not committing changes. (Col. 6, ll 58 – 67 and Col. 7, ll. 27-31). However, removal of data from data cache 206 is not dependent on changes to other data or other system events. Without cache dependencies, there is no need for an extensible cache dependency base class nor any basis for deriving a customized cache dependency base class from an extensible cache dependency base class.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at 801-533-9800.

Dated this 6th day of December, 2007.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
MICHAEL B. DODD
Registration No. 46,437
Attorneys for Applicant
Customer No. 47973

RDN:MBD:crb
CRB0000007026V001